



Warsztaty z programowania w języku Python



Ministerstwo Nauki
i Szkolnictwa Wyższego

Projekt dofinansowany ze środków budżetu państwa, przyznanych przez Ministra Nauki
w ramach programu Społeczna odpowiedzialność nauki II

Nr umowy POPUL/SP/0194/2023/01 z dnia 10.08.2023, kwota dofinansowania 125 730 zł, całkowita wartość projektu 139 700 zł.

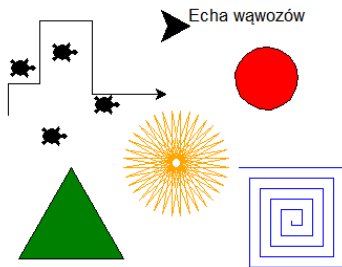


W ramach warsztatów korzystać będziemy ze zintegrowanego środowiska programistycznego **PyCharm** firmy JetBrains dla języka programowania Python

<https://www.jetbrains.com/pycharm/>



Moduł `turtle` w języku Python służy do programowania prostej grafiki, którą tworzy żółwik poruszający się w oknie planszy.





Aby pracować z modułem turtle zaimportuj wszystkie elementy z tego modułu:

```
from turtle import *
```

Teraz możesz rysować kształty: strzałki, koła, kwadraty, trójkąty albo żółwiki. Aby wybrać odpowiedni kształt, nadać mu kolor i ustawić rozmiar należy użyć funkcji:

```
shape('arrow')  
color('blue')  
shapeseize(2,2,1)
```

Kształt zostanie narysowany na środku planszy.



Uwaga: program wykona kolejne instrukcje i zakończy się.
Aby to zatrzymać i obejrzeć wynik rysowania, na końcu programu
użyj funkcji

`done()`



Dla funkcji `shape` możesz wybrać:

`turtle`, `circle`, `square`, `triangle`, `arrow`, `classic`

Dla funkcji `color` możesz wybrać:

`yellow`, `blue`, `red`, `green`, `black`, `brown`, `pink`, `orange`, ...

lub trójkę liczb – kolor w `rgb`, np. `(0.3, 0.6, 0.1)`

Dla funkcji `shapsize` argumenty określają skalowanie wymiarów kształtu i szerokości obramowania. Domyślnie są równe 1.



Dla funkcji `shape` możesz wybrać:

`turtle`, `circle`, `square`, `triangle`, `arrow`, `classic`

Dla funkcji `color` możesz wybrać:

`yellow`, `blue`, `red`, `green`, `black`, `brown`, `pink`, `orange`, ...

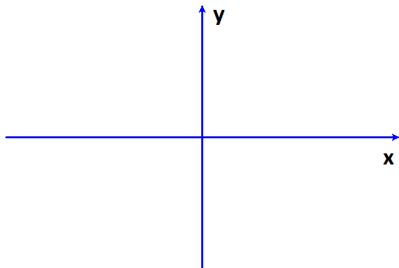
lub trójkę liczb – kolor w `rgb`, np. `(0.3, 0.6, 0.1)`

Dla funkcji `shapsize` argumenty określają skalowanie wymiarów kształtu i szerokości obramowania. Domyślnie są równe 1.

Zadanie: Narysuj zielonego żółwia, ustaw wysokość i szerokość tak, aby zajmował przynajmniej pół planszy.



Układ współrzędnych na planszy wygląda standardowo:



Domyślnie kształt rysowany jest na środku planszy, której rozmiar możesz ustawić funkcją

```
screensize(600,500)
```



Jeśli chcesz zmienić miejsce rysowania, użyj funkcji:

```
goto(150, 60)
```

Uwaga: przy zmianie pozycji rysowana jest linia od punktu startowego do docelowego.

Aby zmienić pozycję nie rysując śladu ruchu, należy "podnieść" pióro przed rysowaniem, a po rysowaniu "opuścić":

```
up()  
goto(100, -60)  
down()
```



Zadanie: Narysuj wybrany kształt w lewej dolnej ćwiartce planszy.

Zadanie: Narysuj kształt w dwóch różnych miejscach planszy...

Okazuje się, że każda instrukcja goto przemieszcza nasz kształt. Aby kształt na stałe pozostał na płaszczyźnie, musimy nim "zrobić pieczętkę":

```
stamp()
```

Dodatkowo możesz obrócić kształt:

```
setheading(90)
```



Zadanie: Narysuj dwa różne kształty w różnych kolorach, obrócone w różne strony w różnych miejscach planszy.



Można również zdefiniować własny kształt:

```
addshape('nowy', ((50, 20), (50, 60), (30, 60), (10, 20)))  
shape('nowy')  
color('brown')
```





Na planszy możesz również pisać:

```
write("Jakiś napis.", font=('Arial', 20, 'normal'))
```



Na planszy możesz również pisać:

```
write("Jakiś napis.", font=('Arial', 20, 'normal'))
```

Zadanie: Napisz swoje imię i nazwisko czcionką o wybranym kolorze i rozmiarze i w dowolnym miejscu planszy.



Jeżeli w programie chcesz używać kilku kształtów zamiennie, każdy powinien być zdefiniowany tak, jak na przykładzie:

```
prostokat = Turtle()  
prostokat.shape('square')  
prostokat.color('red')  
prostokat.shapesize(2,1,1)
```

```
owal = Turtle()  
owal.shape('circle')  
owal.color('blue')  
owal.shapesize(1,2,1)
```





Do obsługi czasu służy moduł `time`.

Zaimportuj ten moduł:

```
import time
```

Aby pobrać aktualny czas, a następnie go wyświetlić użyj instrukcji:

```
czas = time.localtime()  
write(time.strftime("%X", t))
```



Do obsługi czasu służy moduł `time`.

Zaimportuj ten moduł:

```
import time
```

Aby pobrać aktualny czas, a następnie go wyświetlić użyj instrukcji:

```
czas = time.localtime()  
write(time.strftime("%X", t))
```

Zauważ, że osiągnięty wynik nie jest taki sam, jak na zegarku elektronicznym.



Naszym głównym zadaniem będzie narysowanie zegara:

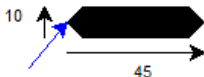
09:07:32



Zadanie: Zbuduj wszystkie cyfry z kształtu kreski zdefiniowanego następująco:

```
addshape("kreska", ((0, 0), (5, 5), (5, 40), (0, 45),  
                    (-5, 40), (-5, 5)))
```

```
kreska = Turtle()  
kreska.shape("kreska")  
kreska.speed(0)  
kreska.up()
```





Zadanie: Zbuduj dwukropki z kształtu kropki zdefiniowanego następująco:

```
addshape("kropka", ((0,0), (0,10), (10, 10), (10, 0)))  
kropka = Turtle()  
kropka.shape("kropka")  
kropka.speed(0)  
kropka.up()
```



Zadanie: Wyznacz współrzędne dla kolejnych cyfr i dwukropków, aby uzyskać następujący wygląd zegara:





Zadanie: Zdefiniuj cyfrę 2

```
def dwa(x, y):  
    kreska.setheading(0)  
    kreska.goto(x + 0, y + 45)  
    kreska.stamp()  
    kreska.goto(x + 0, y + 0)  
    kreska.stamp()  
    kreska.goto(x + 0, y + 90)  
    kreska.stamp()  
    kreska.setheading(90)  
    kreska.goto(x + 45, y + 45)  
    kreska.stamp()  
    kreska.goto(x + 0, y + 0)  
    kreska.stamp()
```



Zadanie: Analogicznie zdefiniuj pozostałe cyfry



Aby pobrać godziny, minuty i sekundy z czasu aktualnego,
użyj instrukcji:

```
czas = time.localtime()  
godziny = czas.tm_hour  
minuty = czas.tm_min  
sekundy = czas.tm_sec
```



Aby wyznaczyć cyfrę jedności i cyfrę dziesiątek liczby dwucyfrowej, użyj instrukcji:

```
cyfra_jednosci = liczba % 10  
cyfra_dziesiatek = liczba // 10
```



Aby wyznaczyć cyfrę jedności i cyfrę dziesiątek liczby dwucyfrowej, użyj instrukcji:

```
cyfra_jednosci = liczba % 10  
cyfra_dziesiatek = liczba // 10
```

Zadanie: Wyznacz kolejne cyfry do wyświetlenia na zegarku.



Aby cyfra była rysowana we właściwym miejscu planszy,
czyli w punkcie (x,y) , zdefiniuj funkcję:

```
def rysuj(cyfra, x, y):  
    if cyfra == 0:  
        zero(x, y)  
    elif cyfra == 1:  
        jeden(x, y)  
    elif cyfra == 2:  
        dwa(x, y)  
    ...
```



Aby cyfra była rysowana we właściwym miejscu planszy, czyli w punkcie (x,y) , zdefiniuj funkcję:

```
def rysuj(cyfra, x, y):  
    if cyfra == 0:  
        zero(x, y)  
    elif cyfra == 1:  
        jeden(x, y)  
    elif cyfra == 2:  
        dwa(x, y)  
    ...
```

Zadanie: Wyświetl wyznaczone cyfry w odpowiednich miejscach planszy.



**DOFINANSOWANO
ZE ŚRODKÓW
BUDŻETU PAŃSTWA**

W RAMACH PROGRAMU
Społeczna odpowiedzialność nauki II
projekt: Matematyczne echa lessowych
wąwozów

DOFINANSOWANIE
[125 730 zł]

CAŁKOWITA WARTOŚĆ
[139 700 zł]

DATA PODPISANIA UMOWY
[08.2023]